

# IBM PC机的指令系统

♥ 8086/8088指令系统分成下列六大类：

- ❖ 数据传送指令
- ❖ 算术运算指令
- ❖ **逻辑运算和移位指令**
- ❖ 控制转移指令
- ❖ CPU控制指令
- ❖ 串操作指令



# 逻辑运算指令

逻辑与指令: **AND DST, SRC**

执行操作:  $(DST) \leftarrow (DST) \wedge (SRC)$

用途: 用于屏蔽一个数的某些位。

逻辑或指令: **OR DST, SRC**

执行操作:  $(DST) \leftarrow (DST) \vee (SRC)$

用途: 用于置位一个数的某些位。

异或指令: **XOR DST, SRC**

执行操作:  $(DST) \leftarrow (DST) \vee (SRC)$

用途: 将一个数的某些位取反。

测试指令: **TEST OPR1, OPR2**

执行操作:  $(OPR1) \wedge (OPR2)$

用途: 用于测试一个数的某些位。

**CF OF SF ZF PF AF**

**0 0 \* \* \* 无定义**

**根据运算结果设置**



# 逻辑运算指令

ASM

逻辑非指令: NOT OPR

执行操作:  $(\text{OPR}) \leftarrow \neg (\text{OPR})$

功能: 按位取反

\* OPR不能为立即数

\* 不影响标志位



# AND VS TEST

♥ AND具有破坏性，TEST没有

❖  $AL = 0FFH$

❖  $AND\ AL, 0$

❖  $TEST\ AL, 0$

♥ 同SUB和CMP



# 例子

例：屏蔽AL的0、1两位  
**AND AL, 0FCH**

```

      * * * * *
AND  1 1 1 1 1 1 0 0
      * * * * *

```

例：置AL的第5位为1  
**OR AL, 20H**

```

      * * * * *
OR   0 0 1 0 0 0 0 0
      * * 1 * * * *

```

例：使AL的0、1位变反  
**XOR AL, 3**

```

      * * * * * * *
XOR  0 0 0 0 0 0 1 1

```

例：测试某些位是0是1  
**TEST AL, 1**  
**JZ EVEN**

```

      * * * * * * *

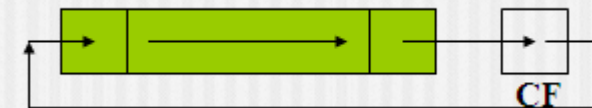
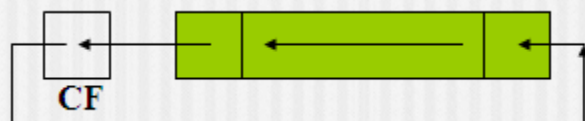
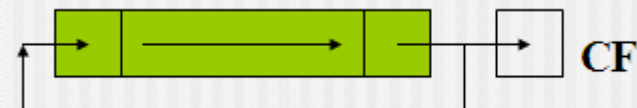
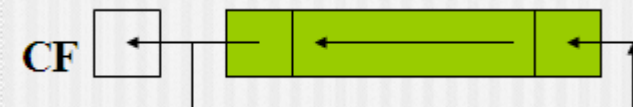
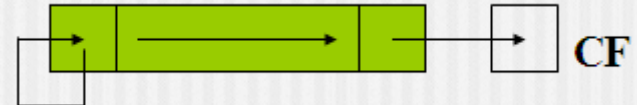
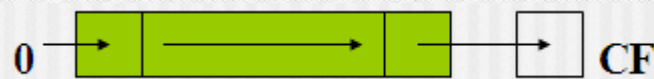
```



# 移位指令

♥ 分类:

- ❖ 逻辑SHL/SHR 算术SAL/SAR
- ❖ 循环ROL/ROR 带进位循环RCL/RCR





# 移位指令

## ♥ 共同特点

- ❖ 都是按位进行
- ❖ 当移动的位数为一位时，用立即数1；当移动二位或二位以上时，要预先将移动的位数存放在CL中。
- ❖ `SHL AL, 2`  $\rightarrow$  `MOV CL, 2 ; SHL AL, CL;`

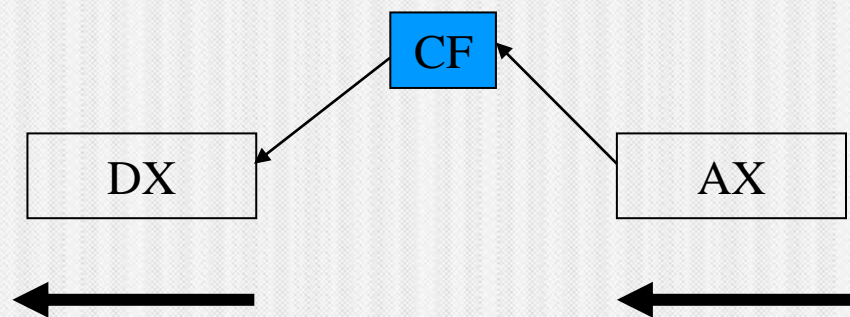


## 例子（一）

♥ RCL和RCR常用在多字节数的移位。

- ❖ 在DX和AX中存放着一个32位数据，试将其左移1位。
- ❖ SHL AX,1
- ❖ RCL DX,1

❖ 右移如何处理？





## 例子（二）

♥ 把(BL)中的8位数高低4位互换

- ❖ MOV CL,4
- ❖ ROL/ROR BL, CL
- ❖ MOV DL,BL
- ❖ MOV CL,4
- ❖ SHR BL, CL
- ❖ SHL DL, CL
- ❖ OR BL, DL